EXHIBIT P

Exhibit 8 – Shirazi

'156 Patent

Claim Limitation (Claim 7)	Exemplary Disclosure
[**156a**] A device comprising:	Shirazi et al.'s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, discloses an example device. *See, e.g.,*: "Many algorithms rely on floating point arithmetic for the dynamic range of representations and require millions of calculations per second. Such computationally intensive algorithms are candidates for acceleration using custom computing machines (CCMs) being tailored for the application. Unfortunately, floating point operators require excessive area (or time) for conventional implementations. Instead, custom formats, derived for individual applications, are feasible on CCMs, and can be implemented on a fraction of a single FPGA." Shirazi et al., *Quantitative Analysis of Floating Point Arithmetic*, at 1.
[**156b**] at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,	Shirazi et al.'s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, discloses at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See, e.g.,*: "Eighteen and sixteen bit floating point adders/subtracters, multipliers, and dividers have been synthesized for Xilinx 4010 FPGAs." Shirazi et al., *Quantitative Analysis of Floating Point Arithmetic*, at 1. "Floating point multiplication is much like integer multiplication. Because floating-point numbers are stored in sign-magnitude form, the multiplier needs only to deal with unsigned integer numbers and normalization. Like the architecture of the floating-point adder, the floating point multiplier unit is a three stage pipeline that produces a result on every clock cycle. The bottleneck of this design was the integer multiplier. Four different methods were used to optimize the integer multiplier in order to meet speed and size requirements." Shirazi et al., *Quantitative Analysis of Floating Point Arithmetic*, at 5; *see also id.* at Fig. 5.

Exhibit 8 – Shirazi

Claim Limitation (Claim 7)	Exemplary Disclosure

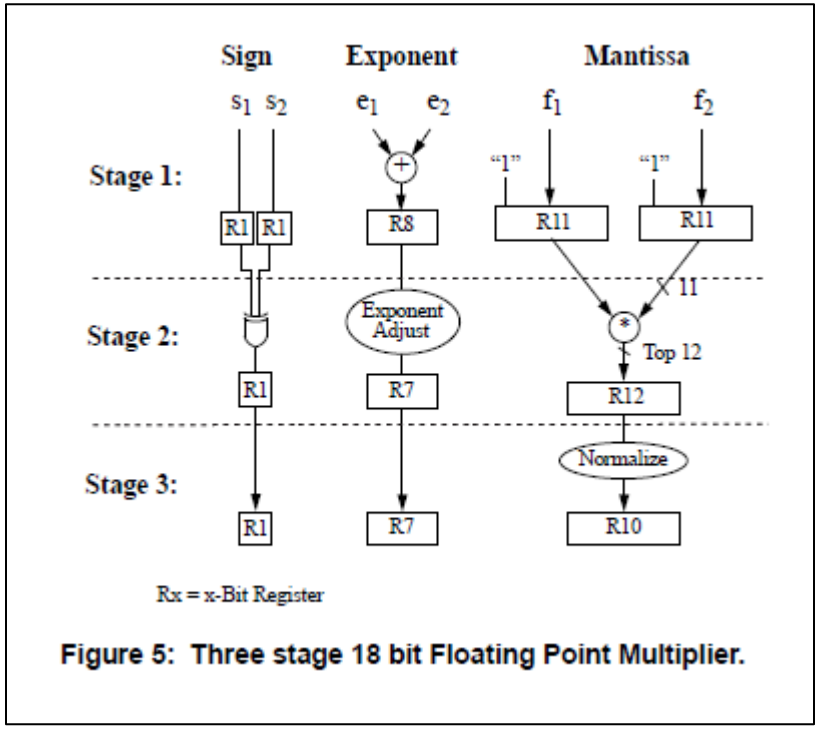


Figure 5: Three stage 18 bit Floating Point Multiplier.

"In order to pipeline the design, three steps prior to the multiplication are necessary: (1) extract the mantissa from the input as the memory address and negate the exponent, (2) provide a delay until the memory data is valid, and (3) insert the new mantissa. . . . The aim in designing the floating point units was to pipeline each unit a sufficient number of times in order to maximize speed and to minimize area It is important to note that once the pipeline is full, a result is output every clock cycle." Shirazi et al., *Quantitative Analysis of Floating Point Arithmetic*, at 6-7.

"Such signal processing techniques necessitate a large dynamic range of numbers. The use of floating point helps to alleviate the underflow and overflow problems often seen in fixed point formats." Shirazi et al., *Quantitative Analysis of Floating Point*

Exhibit 8 – Shirazi

Claim Limitation (Claim 7)	Exemplary Disclosure
	Arithmetic, at 1. "Stage 3: • Normalization of the resulting mantissa is performed. • The resulting sign, exponent and mantissa fields placed into an 18-bit floating point word." Shirazi et al., *Quantitative Analysis of Floating Point Arithmetic*, at 5; *see also id.* at 7 (describing both 16 bit and 18 bit formats),
[**156c**] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; and	Shirazi et al.'s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, discloses the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. *See, e.g.,*: "Such signal processing techniques necessitate a large dynamic range of numbers. The use of floating point helps to alleviate the underflow and overflow problems often seen in fixed point formats. An advantage of using a CCM for floating point implementation is the ability to customize the form." Shirazi et al., *Quantitative Analysis of Floating Point Arithmetic*, at 1. "The second floating point format investigated was a 16-bit representation used by the FIR filter application [7]. Like the FFT application, since multiple arithmetic operations needed to be done on a single chip, we chose a 16-bit format for two reasons: (1) local, 16-bit wide memories were used in pipelined calculations allowing single read cycles only, and (2) more logic was necessary to implement the FIR taps in addition to the two arithmetic units, which do complex number operations. The format was designed as a compromise between data width and a large enough dynamic

Exhibit 8 – Shirazi

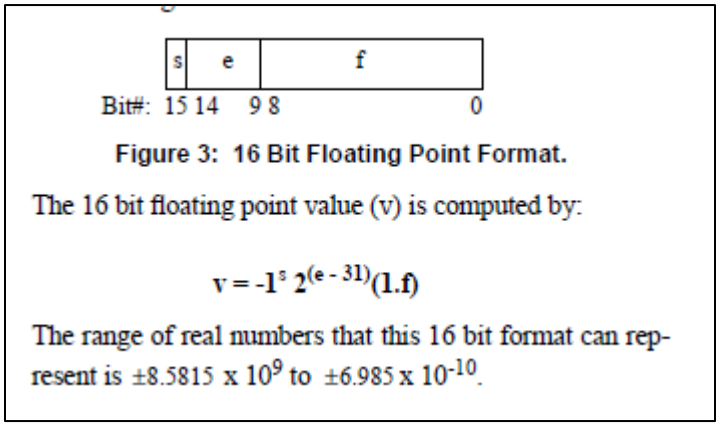
Claim Limitation (Claim 7)	Exemplary Disclosure
	number range. The 16-bit format is shown in Figure 3."  Figure 3: 16 Bit Floating Point Format. The 16 bit floating point value (v) is computed by: $v = -1^{s} 2^{(e-31)}(1.f)$ The range of real numbers that this 16 bit format can represent is $\pm 8.5815 \times 10^{9}$ to $\pm 6.985 \times 10^{-10}$. Shirazi et al., *Quantitative Analysis of Floating Point Arithmetic*, at 2. "To implement single precision floating point arithmetic units on the Splash-2 architecture, the size of the floating point arithmetic units would increase between 2 to 4 times over the 18 bit format. A multiply unit would require two Xilinx 4010 chips and an adder/subtracter unit broken up into four 12-bit multipliers, allocating two per chip[4]. We found that a 16x16 bit multiplier was the largest parallel integer multiplier that could fit into a Xilinx 4010 chip." Shirazi et al., *Quantitative Analysis of Floating Point Arithmetic*, at 7. Figure 3 above depicts a 16-bit case where Shirazi et al. chose a 6-bit exponent and a 9-bit fraction. *See* Appendix to Responsive Contentions Regarding Non-Infringement and Invalidity ("Responsive Contentions") (detailing error rates associated with different mantissa sizes).
[**156d**] at least one first computing device adapted to control the operation of the at least one first	Shirazi et al.'s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, discloses at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit in its

Exhibit 8 – Shirazi

Claim Limitation (Claim 7)	Exemplary Disclosure
LPHDR execution unit;	use of the Splash 2 processing system. *See, e.g.,*: "Many algorithms rely on floating point arithmetic for the dynamic range of representations and require millions of calculations per second. Such computationally intensive algorithms are candidates for acceleration using custom computing machines (CCMs) being tailored for the application. Unfortunately, floating point operators require excessive area (or time) for conventional implementations. Instead, custom formats, derived for individual applications, are feasible on CCMs, and can be implemented on a fraction of a single FPGA." Shirazi et al., *Quantitative Analysis of Floating Point Arithmetic*, at 1. One of skill in the art would have known that the Splash 2 interface contains a controller. *See* Arnold et al., *Splash 2* at 140-41 (describing the DMA controller).
[**156e**] wherein the at least one first computing device comprises at least one of a central processing unit (CPU), a graphics processing unit (GPU), a field programmable gate array (FPGA), a microcode-based processor, a hardware sequencer, and a state machine;	Shirazi et al.'s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, which describes the use of the Splash 2 processor system with LPHDR floating-point arithmetic, discloses at least one first computing device that comprises at least one processing unit and a field programmable gate array (FPGA). *See, e.g.,*: "Many algorithms rely on floating point arithmetic for the dynamic range of representations and require millions of calculations per second. Such computationally intensive algorithms are candidates for acceleration using custom computing machines (CCMs) being tailored for the application. Unfortunately, floating point operators require excessive area (or time) for conventional implementations. Instead, custom formats, derived for individual applications, are feasible on CCMs, and can be implemented on a fraction of a single FPGA." Shirazi et al., *Quantitative Analysis of Floating Point Arithmetic*, at 1. "Although low level design specifications were alternately possible, the strategy used in the work presented here was to specify every aspect of the design in VHDL and rely on automated synthesis to generate the FPGA mapping. . . . The arithmetic operators

Exhibit 8 – Shirazi

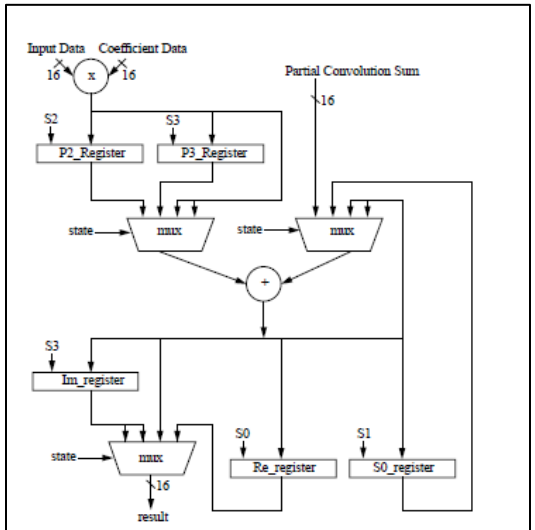
Claim Limitation (Claim 7)	Exemplary Disclosure
	presented here were implemented for real-time signal processing on the Splash-2 CCM, which include a 2-D fast Fourier transform (FFT) and a systolic array implementation of a FIR filter." Shirazi et al., *Quantitative Analysis of Floating Point Arithmetic*, at 1; *see also* Arnold et al., *Splash 2* at 140-41 (describing the Splash 2 processor system for Sun SPARC 2 workstations). "Since the floating point representation already has its fields segregated, the task becomes trivial for a processing element which is complemented by a memory bank of at least 2^n x *n* bits, where *n* is the size of the mantissa's normalized binary representation. Local memories to the processing elements store the reciprocal of each bit combination of the mantissa." Shirazi et al., *Quantitative Analysis of Floating Point Arithmetic*, at 6.  Figure 7: The diagram shows a single Splash-2 PE design for an FIR tap to accomplish complex multiplication. The architecture can achieve two floating-point calculations per clock cycle.

Exhibit 8 – Shirazi

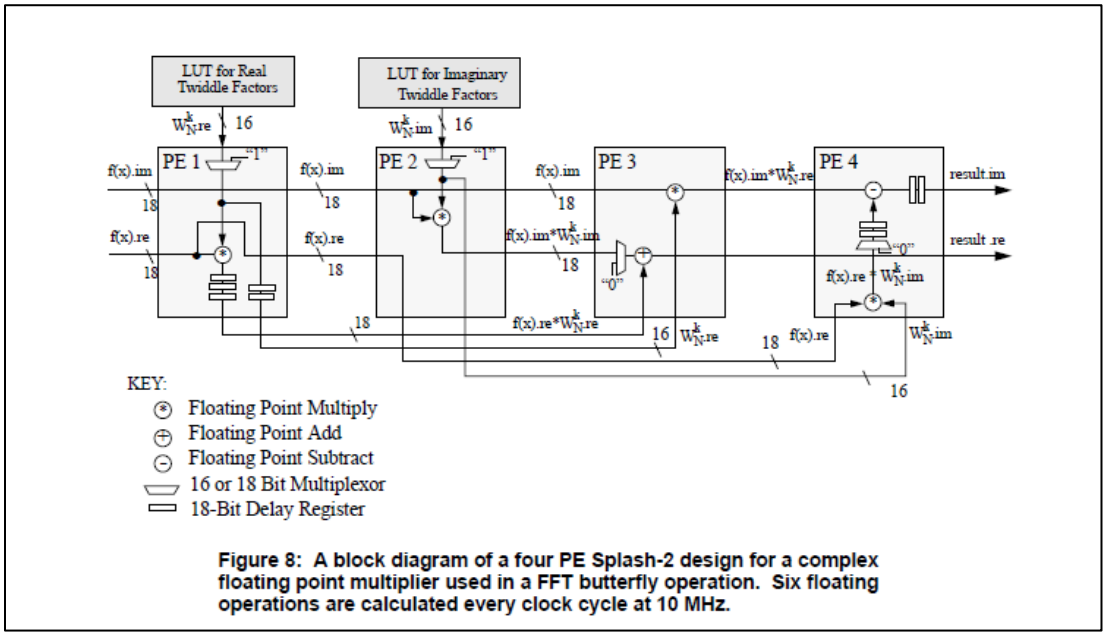
Claim Limitation (Claim 7)	Exemplary Disclosure
	Shirazi et al., *Quantitative Analysis of Floating Point Arithmetic*, at 7. "Each of the floating point arithmetic units has been incorporated into two applications: a 2-D FFT [6] and a FIR filter [7]. The FFT application operates at 10 MHz and the results of the transform are stored in memory on the Splash-2 array board. These results were checked by doing the same transform on a SPARC workstation. An FIR tap design using a floating point adder and multiplier unit is shown in Figure 7. The complex floating point multiplier used in the 2-D FFT butterfly calculation is shown in Figure 8." Shirazi et al., *Quantitative Analysis of Floating Point Arithmetic*, at 8; *see* Fig. 8 (below).  **Figure 8: A block diagram of a four PE Splash-2 design for a complex floating point multiplier used in a FFT butterfly operation. Six floating operations are calculated every clock cycle at 10 MHz.** One of skill in the art would have understood the Splash 2 would be connected to the SPARC 2 workstation, which would contain processors. To the extent Singular contends Shirazi did not disclose at least a microcode-based processor or hardware

Exhibit 8 – Shirazi

Claim Limitation (Claim 7)	Exemplary Disclosure
	sequencer, those would have been obvious elements of computing devices adapted to control the operation of the at least one first LPHDR execution units.
[**156f**] and, wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.	Shirazi et al.'s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, teaches or suggests the possibility of a device wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide. *See, e.g.,*: "Floating point multiplication is much like integer multiplication. Because floating-point numbers are stored in sign-magnitude form, the multiplier needs only to deal with unsigned integer numbers and normalization. Like the architecture of the floating point adder, the floating point multiplier unit is a three stage pipeline that produces a result on every clock cycle." Shirazi et al., *Quantitative Analysis of Floating Point Arithmetic*, at 5.

Exhibit 8 – Shirazi

Claim Limitation (Claim 7)	Exemplary Disclosure

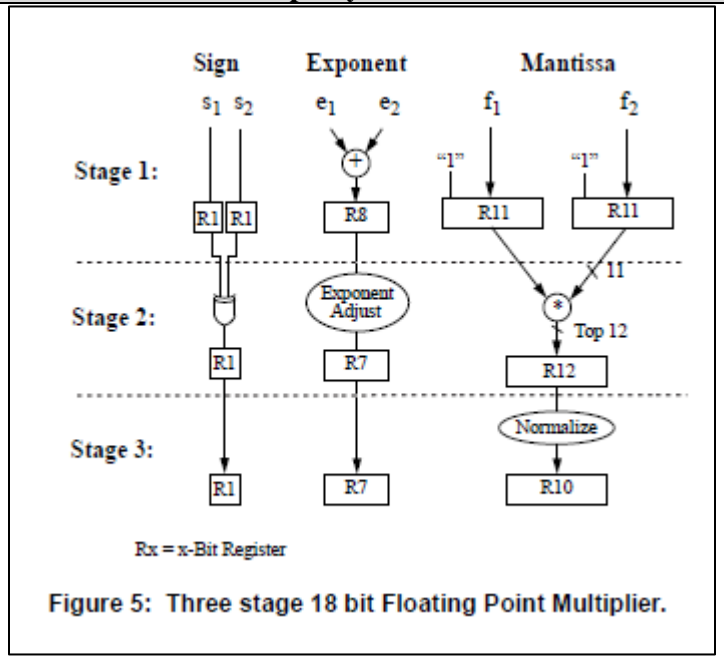


Figure 5: Three stage 18 bit Floating Point Multiplier.

"To implement single precision floating point arithmetic units on the Splash-2 architecture, the size of the floating point arithmetic units would increase between 2 to 4 times over the 18 bit format. A multiply unit would require two Xilinx 4010 chips and an adder/subtracter unit broken up into four 12-bit multipliers, allocating two per chip[4]. We found that a 16x16 bit multiplier was the largest parallel integer multiplier that could fit into a Xilinx 4010 chip." Shirazi et al., *Quantitative Analysis of Floating Point Arithmetic*, at 7.

For the reasons explained in the Responsive Contentions, one of skill in the art applying these teachings of Shirazi would have found it obvious to implement ever greater numbers of LPHDR execution units given intervening developments in FPGA technology and manufacturing.

Exhibit 8 – Shirazi

’273 Patent

Claim Limitation (Claim 53)	Exemplary Disclosure
[**273a**] A device:	Shirazi et al.’s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, discloses an example device. *See* [**156a**].
[**273b**] comprising at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,	Shirazi et al.’s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, discloses at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See* [**156b**].
[**273c**] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X % of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input;	Shirazi et al.’s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, discloses the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. *See* [**156c**].
[**273d**] wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.	Shirazi et al.’s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, teaches or suggests the possibility of a device wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide. *See* [**156f**].

Exhibit 8 – Shirazi

'961 Patent

Claim Limitation (Claim 4)	Exemplary Disclosure
[**961a**] A device comprising:	Shirazi et al.'s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, discloses an example device. *See* [**156a**].
[**961b**] at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,	Shirazi et al.'s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, discloses at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See* [**156b**].
[**961c**] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=10% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.2% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; and	Shirazi et al.'s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, discloses the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. *See* [**156c**].
[**961d**] at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit.	Shirazi et al.'s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, discloses at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit in its use of the Splash 2 processing system. *See* [**156d**].

Exhibit 8 – Shirazi

Claim Limitation (Claim 13)	Exemplary Disclosure
[**961e**] A device comprising:	Shirazi et al.'s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, discloses an example device. *See* [**156a**].
[**961f**] a plurality of components comprising:	Shirazi et al.'s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, discloses at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See* [**156b**]. Shirazi et al.'s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, discloses at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit in its use of the Splash 2 processing system. *See* [**156d**].
[**961g**] at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,	Shirazi et al.'s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, discloses at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See* [**156b**].
[**961h**] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=10% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.2% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input.	Shirazi et al.'s article, *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines*, discloses the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. *See* [**156c**].